## Section 2 – Configuration

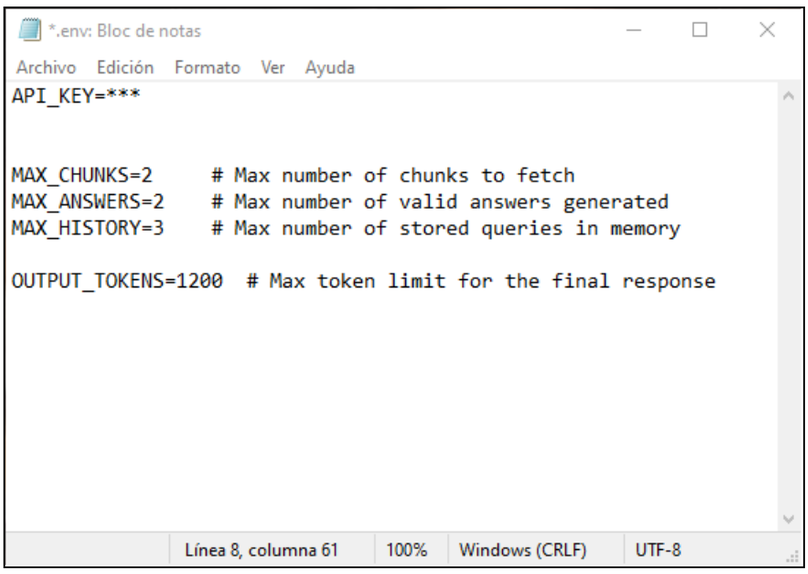
This English demo version comes in one version:

Versión - conceptCurve_english_UCC.zip: This contains the Uniform Commercial Code of the State of Michigan, a document of approximately 220,000 tokens, divided into 17 chunks.

To see how the software works, you can check video number 6 on youtube channel:

https://www.youtube.com/@Agente_Concept_Curve

### Software Configuration




In the root directory where the software has been installed, you will find the configuration file. The configuration options are:

(1) API_KEY To operate the software, the user must subscribe to an API Key. Initially, the software is designed to work with OpenAI, but it can be adapted to other AI providers, including offline options. Developers can customize this by modifying the smartFunctions.js file.

(2) MAX_CHUNKS Determines the number of chunks the software will search through to answer the user's query.

Example: For the Uniform Commercial Code, a document divided into 17 chunks, it's good to configure: MAX_CHUNKS=2 or 3

(3) MAX_ANSWERS Sets the maximum number of relevant answers the software will find before stopping the search. A recommended practice is to leave this at 2, but it depends on the document and the query involved.

Example: If we consult the UCC about mentions of "good faith," we will likely get many positive results compared to asking a more narrow and specific question.

(4) MAX_HISTORY The software does not store the entire conversation history in memory as this would be inefficient. It keeps a minimal history that is enough to maintain conversational context. Each query uses this limited context for continuity.

In practice, MAX_HISTORY=3 is sufficient in all cases.

A trick the software uses to avoid unnecessary computational costs is through a **Smart-Function** that refines the question, creating an ideal prompt for the AI to maintain conversational flow. That's why the full history isn't necessary; MAX_HISTORY=3 suffices.

**What are Smart-Functions?** This concept was developed by Daniel Bistman. More info is available in **Section 3 – For Developers**.

You can also ask the CC Agent ---> https://tinyurl.com/agent-cc

(5) OUTPUT_TOKENS This sets the maximum length the software will use to deliver the polished final answer presented in the frontend.